Software Project Management Plan

Team Aura
Virtual Surgery

May 17, 2011 *Revision* : 1.42

Abstract

This document formally outlines the management procedures to be used by Team Aura during the Virtual Surgery Project. It details team structure, project organisation, managerial and technical processes, and the project schedule.

Contents

1 Introduction

This document is Team Aura's Software Project Management Plan for the Virtual Surgery project. The intended audience for this document is all team members, and the project supervisors.

This document is not intended to stand alone. It should be read in conjunction with the other documents produced for this project, as listed in Section 1.5 of this document. This is to avoid duplication of information between documents. A pointer to the project's acronym file is given in section 1.6.

This document is based on IEEE Standard 1058.1-1987 for Software Project Management Plans. The sectional layout is as specified, except for two deviations:

- This section contains an extra subsection, 1.3 Resource Requirements and Availability;
- The format of section 5 Project Schedule is different; it has been changed to suit the nature of this project more appropriately.

Note: Some sections are extremely short and only reference other sections and documents. They are present to comply with the IEEE standard.

1.1 Project Overview

The Virtual Surgery project is being undertaken by Team Aura for the subject 433-440 Advanced Software Engineering Project. This subject is compulsory for all final year software engineering students at the University of Melbourne.

1.1.1 Project Description

Virtual Surgery is the code name for a general purpose surgery simulation software system which will enable surgeons to practice general mastoid surgery in a virtual environment.

It is a software system which will realistically render CT scans of the skull and allows surgeons to interact with this virtual skull by removing bone by means of a virtual burr. To increase the realism of the experience, the software system will also provide features such as sound and zooming.

The Virtual Surgery software system provides a safe and cost effective way for surgeons to learn the fundamental techniques of general mastoid surgery. By doing away with the need for physical material, the software system allow the surgeons to improve their skills by practising more frequently in an easily accessible and reproducible environment.

The Virtual Surgery software system is a step towards the advanced training of surgeons in surgical intervention which remains a key element in improving the control of hearing and balance disorders and enhancing the health of many individuals.

The task to build the Virtual Surgery software system is being undertaken by Team Aura as part of *433-440 Advanced Software Engineering Project*.

1.1.2 People

The members of Team Aura are listed in Table 1.

The clients for this project are listed in Table 2. Both clients are surgeons at the Royal Victorian Eye and Ear Hospital.

The supervisors for this project are given in Table 3.

Name	*login*	*Phone*
Omitted to preserve anonymity		

Table 1: Team Aura Members

Name	*Email*	*Phone*
Omitted to preserve anonymity		

Table 2: Clients

Name	*login*
Omitted to preserve anonymity	

Table 3: Supervisors

1.2 Project Deliverables

The project deliverables are:

- Copies of documents produced during course of the project, as listed in Section 2.1 of the SQAP;
- A copy of the source code produced;
- A copy of a running version of the software as derived from the source code, plus any extra files or libraries required;
- User documentation as described in the SRS.

These will be given to the clients at the completion of the project, on or soon after October 20, 2000. All deliverables will be given to the clients on a CD-ROM.

The team's group directory will be left in a state so that markers can easily access each of the deliverables. Any hardcopy-only deliverables, such as parts of the design notebook, will be physically given to the project supervisors.

1.3 Resource Requirements and Availability

Team Aura has limited access to resources, such as physical resources and personnel resources during the course of the project. The budget for this project is limited by university funding as well. This section is not part of the IEEE Standard 1058.1-1987.

1.3.1 Physical Resources

Physical resources available to the team include:

- Computers in the 4th year lab, room L2.03, reserved for 4th year students;
- Computers in the basement labs;
- Any software available on any of these computers;
- Room L2.04, the 4th year lounge, is available for meetings;
- The Engineering library at the University of Melbourne, and any other libraries to borrow books, articles and journals;
- The cabinets and drawers in room L2.04 provided for the storage of project related materials.

1.3.2 Personnel Resources

Personnel resources available, outside the team itself, include:

- The project supervisors, are able to provide general help and advice when necessary;
- Department staff may be able to provide help and information on technical matters.
- The department's system administrators and technical services employees can provide general hardware, software and network support, to a reasonable extent consistent with the inherent responsibilities of their positions;
- The local newsgroup `cs.chat` can be used to access the collective wisdom of gurus within the department on specific matters;
- Any appropriate non-local newsgroups can be used as a means of research into specific technical matters.

1.3.3 Budget

As this project is being undertaken as a university subject, no formal budget is allocated. Some small amounts of funding may be provided by the Department of Computer Science and Software Engineering, e.g. for any necessary software or hardware (such as the GeForce card). The project supervisor will be contacted should any such funding be required.

1.4 Evolution of the SPMP

The Project Manager worked on this document freely during initial drafting. It then underwent informal, formal, and two external reviews (one by teams Centaur and 3dis2, one by the project supervisor). Substantial changes were made in response to these reviews. The document is being continually updated as management procedures are refined.

The SPMP will not be baselined due to the evolving nature of the document. Please refer to the SCMP for further details on which other documents are not baselined.

The Project Schedule will be updated approximately weekly as the project proceeds (see section 5 for more details). Please refer to the SCMP for the location of the project schedule

1.5 Reference Materials

The SPMP refers to the following documents:

- Software Quality Assurance Plan (SQAP);
- Software Configuration Management Plan (SCMP);
- Risk Mitigation, Monitoring & Management Plan (RMMMP);
- Project Schedule.

Other references used in the creation of the document include:

- IEEE Standard 1058.1-1987;
- Team KISS, SPMP, 1999;
- Team Sphere, SPMP, 1999.
- 433-443 Software Project Management lecture notes, Ricardo Fiusco, 2000.

1.6 Definitions and Acronyms

A comprehensive list of all acronyms used in the project is available in the team's CVS repository, under: `/VirtualSurgery/doc/resources/acronyms`. Details on the structure of the repository are given in the SCMP.

2 Project Organisation

2.1 Process Model

The development model chosen for the project is a prototype-driven incremental model, using the terminology given in the 433-443 Software Project Management lecture notes. Please refer to these notes for more details.

The stages of this process are described in the following sections.

2.1.1 Requirements Gathering

A standard requirements elicitation process will be used, based primarily on weekly meetings with the client during the requirements phase.

Requirements are being partitioned into core and non-core; the client will be asked to rank non-core requirements in order of desirability. The core requirements define the functionality which will be developed in increment one. The non-core requirements are those which are to be implemented in increment two.

The requirements gathering phase will culminate in the baselining of the SRS, which will be signed-off by the clients.

2.1.2 Prototyping

Standard requirements gathering and modelling is being accompanied by the development of a throwaway prototype. This approach has been chosen to address two main risks:

1. The clients are somewhat vague about the exact functional requirements for the system, and the kind of hardware the system needs to run on;
2. The technical feasibility of the requirements is open to question, and it is unclear which of several possible approaches would be most appropriate to give the level of performance required.

Creating and evaluating a prototype should be an excellent method of mitigating these risks. The specific issues the prototype is intended to address are:

- User interface;
- Viability of various 3D rendering techniques;
- Possible methods of 3D model interaction and update.

2.1.3 Architectural Design

The SADD work shall begin after the prototyping has finished. The prototype as mentioned above, is intended to be a throwaway prototype and therefore not to be developed into the final system. The architectural issues addressed by the prototype subteam, and the experience and insight gained by them will be put to best use by including members of the prototype subteam in the architectural design subteam.

Architectural design will cover both core and non-core requirements, as it is essential that the structure of the overall system be mapped out at the beginning of the design phase.

2.1.4 Increments One and Two

The core requirements of the system will be implemented in increment one. The non-core requirements are to be implemented in increment two. The second increment will follow a two-build process. Each build will consist of developing certain non-core requirements, depending on the prioritisation as specified by the clients. At each build the system will be compiled and system tested. The next build can then begin. Each build can be likened to a mini-increment in itself. The rationale behind using builds can be explained by the nature of our project. Since some of the non-core requirements may be harder to implement, having smaller milestones where the whole code could be compiled and tested would deliver a working product with added functionality. The functionality to be implemented in the two builds are to be based on prioritisation of requirements by the clients.

Each increment will involve:

- Detailed design;
- Detailed test planning;
- Coding;
- Unit, integration, acceptance, and release testing;
- Creation of user documentation.

Increment one will begin once architectural design has been completed. Increment two will begin once increment one system testing has been completed. The user documentation will only be produced during increment two.

2.1.5 Other Details

The relationships between major project functions and activities — such as the timing of major milestones, baselines, reviews, work products, and deliverables — are given in the Project Schedule; see section 5 for details.

2.2 Organisational Structure

2.2.1 Organisation structure

The team will adopt a controlled decentralised organisation model for the project. The team will be split up into subteams and development groups during different phases of the project. Each subteam and development group will have an appointed leader. Subteam and development group leaders will be decided in team meetings. The Management team consists of all the subteam and development group leaders. For more information regarding the management team, refer to section 2.2.2.

Each subteam and development group may vary in size as the project proceeds. This will reflect the changing amount of work that needs to be done for different tasks at each point in time. For example, the requirements subteam will shrink and

then disband once requirements analysis is completed; the design subteam will be non-existent prior to the design phases.

For this reason, each team member may be part of more than one subteam and development group at a time. Membership to a subteam and development group is not intended to restrict any team member to tasks covered by that subteam only. Instead, it should act as a guide to the tasks they should be performing, indicate which other team members they should be in close communication with, and the subteam and development group leader to whom they should report (if applicable).

Further, during the detailed design and implementation phases of the project, the team will be split up into three development groups (DGs) that will independently code and test different parts of the system architecture.

2.2.2 Management Role

We believe that this organisation model will suit the needs for Team Aura. There is sufficient control over the project, which is exercised by the Management team. Subteam and Development Group leaders manage their particular groups and subteams, and inform management team of progress. This ensures that the Management team maintains sufficient control, and at the same time the effort required to designate tasks and allocate resources does not become counterproductive.

There are also crossteam representatives who are responsible for communicating with Team Centaur and Team 3dis2; details regarding these representatives are given in section 2.3.4. A schematic of the team structure is shown in figure 1.

2.2.3 Subteams and Responsibilities

The leaders (marked with a *), members, and responsibilities of the various subteams are:

Management:

- Handle project planning, scheduling, and high level task distribution to both individual, subteams and development groups;
- Write and maintain the SPMP;
- Identify risks as they come up in the course of the project;
- Oversee the creation of risk management procedures and documentation, and to monitor risks;
- To resolve any conflicts.

Team members: Omitted.

Quality Assurance:

- Supervise the major quality assurance activities, such as the defining and enforcing of appropriate standards, and the undertaking of reviews and audits;
- Write and maintain the SQAP;

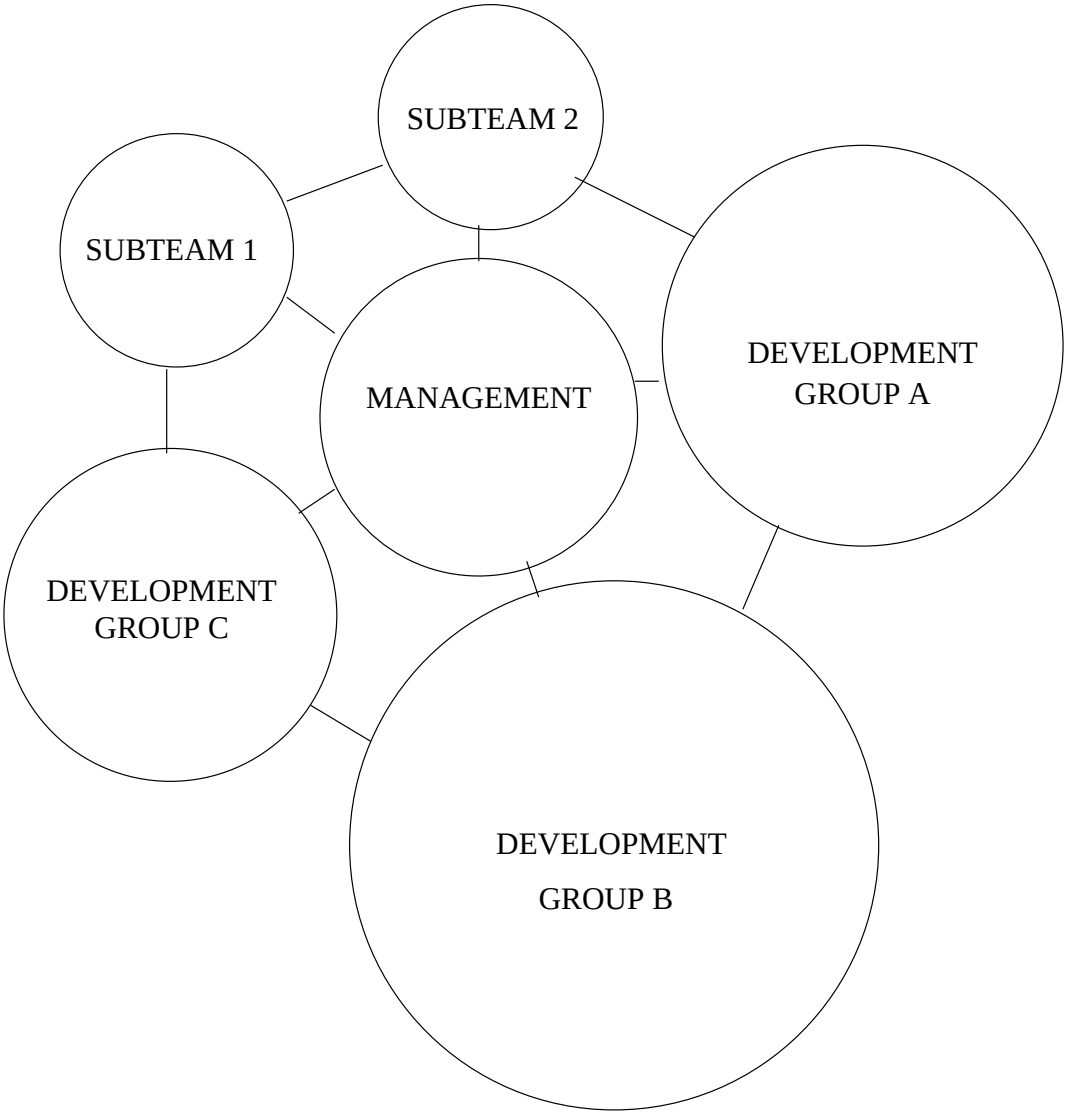

Figure 1: Aura team structure schematic (larger circle indicates more people)

- Write and maintain the SVVP;
- Supervise cross-team QA (e.g. organise external reviews for Centaur and 3dis2 documents);
- Supervise cross-document consistency checking.

Team members: Omitted.

Configuration Management:

- Supervise all aspects of configuration management, such as the definition of appropriate standards;
- Establish and maintain baselines and releases;

- Develop and maintain the traceability database;
- Develop and maintain any specialised SCM tools and support software;
- Perform configuration control;
- Perform configuration status accounting;
- Participate in SQA Audits;
- Write and maintain the SCMP.

Team members: Omitted.

Requirements:

- Initially gather, and then refine requirements by having regular meetings with the clients;
- Undertake requirements modelling, including use cases;
- Split requirements into core and non-core, to be implemented in increments one and two respectively;
- Write and maintain the SRS;
- Provide requirements details and knowledge to the prototype team, should the prototype team need them.

Team members: Omitted.

Prototype:

- Research various methods for rendering and interacting with 3D volumes;
- Evaluate various graphics toolkits and libraries for suitability to the project;
- Perform exploratory programming to determine the feasibility of various functional and non-functional requirements;
- Create a prototype, and evaluate it in conjunction with the clients.

Team members: Omitted.

Training:

- Compile training manuals on development technologies;
- Organise a selection of tutorials on the team web page for team members to learn Java and Java3D over the mid-year break;

Team members: Omitted.

Architectural Design:

- Design the system architecture;
- Document the system architecture in the SADD;
- Perform the initial Translate the SADD over to Javadocs ready for detailed design;
- Support detailed design and implementation activities by answering member queries regarding architectural design.

Team members: Omitted.

Development Groups: Development Groups (DGs) are responsible for their allocated part of the system architecture. There are three DGs: GUI, Simulator and World. Each group has the following roles and responsibilities relevant to their part of the system:

- Produce detailed design, updating and adding detail to that produced by the architectural team as implementation progresses;
- Produce unit and intra-package test cases;
- Implement the design;
- Conduct code inspections on all code produced;
- Conduct unit testing;
- Conduct intra-package integration testing;

GUI Team members: Omitted. *Simulator Team members:* Omitted. *World Team members:* Omitted. During increment two, Simulator and World merged into the Simworld development group to reduce communication overhead.

2.2.4 Individual Roles and Responsibilities

The responsibilities of the individual roles are:

Project Manager:

- Oversee project planning, scheduling, and high level task distribution, with the aid of the management subteam;
- Maintain project schedule up to the beginning of the coding phase;
- Maintain good relationships between team members and between leaders of subteams;
- Conduct regular meetings with subteam leaders;
- Chair all full team meetings according to meeting procedures, or find a substitute chairperson if he cannot attend.

Assistant Project Manager:

- Act as a backup for any time the Project Manager is unable to conduct his duties; in such a case, the Assistant Project Manager will assume the same responsibilities as the Project Manager;
- Remain aware of the Project Manager's affairs, so she can assume the Project Manager's role with a minimum of effort, if necessary.

Configuration Manager:

- Manage the team's group directory;
- Manage the team's CVS repository and ensure it follows CM standards;
- Maintain the SCMP in conjunction with the Quality Assurance Manager and the CM subteam.

General Secretary:

- Take minutes of all full team meetings and commit them to the repository, or arrange for them to be taken and committed by someone else.

Client Liaison Officer:

- Act as first point of contact for client communication;
- Organise client meetings;
- Prepare an agenda for client meetings, or arrange for an agenda to be written;
- Report back to members not present at client meetings;
- Ensure that minutes are taken at all client meetings.

Risk Manager:

- Oversee the creation of the initial risk table;
- Maintain awareness of risks via the risk monitoring procedures;
- Maintain an up to date risk table representing current state of the project;
- File regular reports describing the top ten risks facing the team;
- Identify when any particular risk has become an event, and bring appropriate risk management procedures (as defined in the RMMMP) into play to minimise the impact of the event;
- Poll the rest of the team at each meeting if they feel a risk may arise;
- Collect continual and up-to-date feedback from the the Project Manager and subteam leaders to gauge the project's progress and monitor risks;
- Create and maintain the RMMMP in conjunction with the Quality Assurance Manager and the management subteam.

Quality Assurance Manager:

- Ensure the SQAP remains up to date and make modifications as needed;
- Ensure that all documents are kept up to date, in conjunction with the appropriate team members;
- Ensure that the team is familiar with procedures and standards outlined in the SQAP, SCMP and SPMP;
- Co-ordinate regular audits and reviews in accordance with the review procedures outlined in the SQAP and SCMP;
- Ensure that reviews and audits are followed-up;
- Liaise with other 440 teams in order to share some QA responsibilities.

Technical Leader:

- Be the first port of call for team members experiencing technical difficulties;
- Undertake research into technical questions if necessary;
- Maintain project schedule during increments one and two.

Test Manager:

- Coordinate the testing activities of Team Aura;
- Manage the Test Plan;
- Be the first port of call for team members with any questions regarding testing;
- Ensure all testing activities are carried out in accordance with the Test Plan.

Cross-team Technical Liaison Officer:

- Represent Team Aura in the administration of the computers in lab L2.04;
- Communicate with the Inter-team Technical Liaison Officers from Team Centaur and Team 3dis2 when necessary;
- Administer the team's email archive and the archive's web interface.

Performance Analyst:

- Manage the creation and updating of the SPEAP;
- Collate performance reports, miscellaneous findings and research material into the SPEAP;
- Analyse performance reports and provide the best course of approach to handle performance concerns;
- Inform the Risk Manager of performance issues becoming risk issues.

Usability Analyst:

- Create a usability plan;
- Conduct heuristic evaluation and user testing;
- Collate results and identify severity ratings;
- Make recommendations to the design team based on the results compiled.

Subteam Leader

- Account for and allocate tasks related to the subteam;
- Identify any issues that require clarification, and communicate with team management to resolve any confusion;
- Help maintain and monitor changes to any documents produced;
- Co-ordinate with the Quality Assurance Leader to schedule reviews and ensure procedures are adequate and being followed;
- Assist the Project Manager in project planning for phases of the project in which the subteam is involved;
- Communicate regularly with the Project Manager and other subteam leaders so everyone is aware of the project's progress;
- Ensure that adequate records are kept of team decisions, allocated tasks and their predicted and actual completion dates.

Development Group Leader

- Manage the development of a single package in the system architecture;
- Account for and allocate tasks related to the development group;
- Assist the Project Manager and Management Team in project planning for the two increments;
- Work with the Test Manager to ensure all testing is carried out according to the Test Plan;
- Work with the Technical Leader to ensure all technical aspects of the package are adequately covered;
- Work with the Risk Manager to ensure any identified risks are actively mitigated and managed effectively;
- Ensure the features, testing requirements and detailed design for each increment and build is delivered by the specified deadlines;
- Ensure all team members in the development group work effectively and cooperatively;
- Ensure that adequate records are kept of team decisions, allocated tasks and their predicted and actual completion dates;

All Managers

- Know what your goal and role is;
- Know what your deadline is;
- Estimate the effort required to achieve your goal (e.g. in person-hours);
- Plan a schedule consistent your deadline and the overall Project Plan;
- Request team resources based upon your effort estimation
- Request resources in consultation with team members and their DG Leader;
- Confirm with people before assigning them a task;
- Ensure people know what you expect them to do;
- Ensure people are able to do the best job they can by providing information, resources and support;
- Supervise tasks with progress reports and help when required;
- Ensure you contribute to team tasks as well as delegating;
- Ensure all team standards and processes are adhered to;
- Give due recognition for work completed;
- Make sure people know when they are under-performing;
- Be firm, but fair.

Team Member

- Contribute ideas and opinions in all phases of the project;
- Perform all assigned tasks to the best of their ability and in a timely manner;
- Be prepared to assist other team members when the need arises;
- Spend on average 12–15 hours per week on the project;
- Be aware of risks to the project and alert management of any new risks;
- Report any anomalies regarding the project to management through processes outlined in SQAP, SCMP and SVVP;
- Work together with other team members to avoid conflict;
- Follow all defined standards and procedures.

2.3 Organisational Boundaries and Interfaces

2.3.1 General Team Communication

For all levels within the team structure, there are two recorded methods team members can use for communication:

1. Meetings: recorded via agendas and minutes;
2. Email: the team's mail filtering system allows emails to be sent to only some team members, while still being archived in a way that allows anyone to look at them later.

Details of the mail filtering system are given in the SQAP.

In addition to these formal methods, team members are strongly encouraged to keep themselves aware of what others are doing by talking, or sending (unrecorded) emails when necessary.

2.3.2 Inter-subteam/Development Group Communication

Communication between subteams will be facilitated primarily by regular subteam leader/management meetings. Relevant details should then be passed on by the subteam leaders to the rest of their subteam.

2.3.3 Intra-subteam/Development Group Communication

The primary avenue of communication within subteams/development groups is subteam meetings.

For keeping track of changes to documents, the CVS repository will be set up so that subteam and development group members will be automatically emailed CVS log messages for all changes made to the artifact(s) that are their responsibility. This ensures all subteam members know what changes have been made, and encourages log messages of higher quality.

Emails will be a secondary avenue of communication within subteams and development groups. Each subteam and development group will be assigned an alias, so that emails will only be circulated through members of that subteam and development group. This ensures that subteam members and development group members are in touch with each other, and know the progress being made by fellow team members. These emails are archived, and therefore any team member may view these. Please refer to SCMP regarding email archives.

See `/home/440/Aura/Repository/CVSROOT/loginfo` for the lists of people emailed when changes are made to different parts of the CVS repository.

More details on CVS can be found in the SCMP.

2.3.4 External Communication

Supervisors: The Project Manager will be the first point of contact for communication between Team Aura and the 440 project supervisors.

Cross Team Cooperation Group: For communication between Team Aura and the other 440 teams, the Cross Team Cooperation Group (CTCG) will be the first point of contact. The CTCG consists of the Project Managers and QA Managers of all three teams and a Cross Team Coordinatorand is the designated forum for the following activities:

- Crossteam (external) reviews and audits;
- Resolution of crossteam conflicts;
- Coordination of shared resources like the 4th Year Lab;
- Sharing of ideas and discussion of problems that affect all teams.

More information is found in the Cross Team Cooperation Document (CTCD).

Other: General reports regarding the progress of the project will be made in the weekly 440 class, held at 2.15pm each Wednesday during semester one. One person from the team will present to the project supervisors and the other 440 students. The Project Manager will usually make the report, but this may vary, depending on the structure and topic of the class, as chosen by the Project Supervisors.

2.4 Project Responsibilities

Individual and subteam responsibilities for project support functions are explained in section 4.3.

Individual and subteam responsibilities for project activities are given in the Project Schedule; see section 5.

3 Managerial Process

3.1 Management Objectives and Priorities

The aims of management are to:

- Deliver a high quality, useful product by the delivery date;
- Follow a clear, structured software engineering process;
- Utilise all personnel and resources efficiently and effectively;
- Ensure that all team members follow software engineering practises and principles.

Any existing software that can increase the quality of the final product shall be used, as long as it is:

- Suitable for the project's needs and constraints;
- Available (i.e. free or within resource constraints);
- Acceptable to the clients.

Due to the inflexibility of the delivery date, the project schedule will assume a higher priority than the system requirements — an incomplete system will be given to the clients rather than extending the project deadline.

3.2 Assumptions, Dependencies and Constraints

It is assumed that there will not be a significant change to the team during the project. The project is not particularly dependent on any external events, and the pro-active approach taken towards risk management should help prevent any extraordinary events overwhelming the project. The most significant constraint binding the project is the inflexibility of the delivery date.

3.3 Risk Management

A wide range of risks have been identified for the project. For details on mitigation, monitoring, and management of these risks, please see the RMMMP. Also refer to the Top 10 Risks and Risk Table. The location of Risk Table and Top 10 risks is outlined in the SCMP.

3.4 Monitoring and Controlling Mechanisms

The communication methods outlined in section 2.3, in particular the subteam leader meetings, will be the primary way in which project progress will be monitored. The four methods described below will also be used.

3.4.1 Timesheets

Each team member is strongly encouraged to fill out an individual timesheet file as a record of the amount of work they have done. This may serve the purpose of keeping an account of the tasks the team member has completed during the course of the project. But no formal reports on work done are to be completed as this level of work monitoring was felt to be excessive for this project.

3.4.2 Task Allocation System

Semester One In semester one, the task allocation system consisted of a suite of Perl scripts that managers used to assign and keep track of tasks. This system is no longer in use and has been replaced by a task log system.

The Task Allocation System was stopped in response to the Management Review conducted at the start of second semester. From the survey, it was found that a task log system would better meet the needs of the team in second semester. [1]

Semester Two In semester two, task logs are used to assign and keep track of tasks.

Each development group has one task log file which is updated at weekly development group meetings. Tasks are assigned with an associated deadline that typically gives a person a few days to complete the task. When the task log is updated, each task is assigned a status such as "Done", "Extended" or "Late". A "Reassigned" status means a task has been reassigned to another person. A "Stop" status indicates the task has been stopped.

The person in charge of assigning tasks is the development group leader who has the responsibility of ensuring the task is done and followed up.

Task logs are located at:

- `/VirtualSurgery/doc/design-notebook/tasklog/`

Task database system For the last 16 days of the project, a central task database was used to assign and keep track of tasks. [2] This resulted from the dismantling of development groups and meant all tasks would be managed by a single manager. This level of micro-management was used to avoid any task slippage during this final critical period.

A task schedule was regularly broadcast to the team via email, with a task matrix indicating the task allocations for each member of the team. The task database and task matrix are located at:

- `/VirtualSurgery/doc/design-notebook/tasklog/`

3.4.3 Audits

Periodic audits organised by the QA subteam will monitor adherence to project standards and procedures. Details of Audits are given in the SQAP.

[1] Please see the survey for more details at: `/VirtualSurgery/doc/RMMMP/surveyreport`

[2] The task database system is a MS Access database started on Wednesday, October 4, 2000 and was used until Friday, October 20, 2000 (the final project deadline).

3.4.4 Surveys

Informal/formal surveys of all team members will be conducted by the Management team. These surveys may take the form of a managerial review. Please refer to the SQAP and SVVP regarding managerial reviews. These will take two forms:

- Email questionnaires, regarding specific, individual-related matters;
- Anonymous written questionnaires at team meetings regarding more general team and management matters.

The contents of the questionnaires will vary depending on what the Management team feel are relevant issues at the time, but will typically cover items such as:

- Effectiveness of team, inter-subteam and intra-subteam communication;
- Appropriateness of workloads;
- Quality of procedures;
- Perceptions of various team member's roles and responsibilities;
- Any observed problems.

The intention of these surveys is to elicit information that team members might be somewhat shy or reluctant to bring up at team meetings. It may also lead to taking action regarding some previously unidentified issues, and may become a QA matter as well.

The results of these surveys will be kept in the directory:

- `/VirtualSurgery/doc/RMMMP/`

3.5 Staffing Plan

3.5.1 Available Staff

The people available to complete the project are exactly the seventeen mentioned in section 1.1.2. The team composition was prescribed entirely by the project supervisors. The team is not expected to change during the course of the project.

3.5.2 Subteam Allocation

The allocation of individuals to the various subteams will be decided by the management subteam. A variety of factors will influence this, including:

- Interest;
- Past experience;
- Ability;
- Group dynamics;
- Past opportunities to do tasks of interest during the project.

Management will endeavour to place team members in areas they want to work on whenever possible.

3.5.3 Training

Any necessary training will be conducted entirely within the team under the auspices of the training subteam.

4 Technical Process

4.1 Methods, Tools and Techniques

The computing systems, development methodology, and programming languages to be used during project development are outlined in the SQAP.

The technical standards, policies and procedures governing the development or modification of project deliverables are specified in the SQAP and the SCMP.

4.2 Software Documentation

The documents that will be produced during the course of the project are described in the SQAP.

Procedures for document modification, reviews, and baselining are outlined in the SQAP and the SCMP.

4.3 Project Support Functions

The following three standard support functions (as defined by the IEEE SPMP standard) will be used during the project:

- Quality Assurance: please see the SQAP for details;
- Configuration Management: please see the SCMP for details;
- Verification & Validation: please see the SVVP for details.

Each of these documents give the appropriate details for their respective support functions, such as responsibilities involved, resource requirements, and relevant schedules.

5 Project Schedule

The Project Schedule contains the planning of major activities to be completed during the course of the project by Team Aura. It is stored as several MS Project files located at:

- `/home/440/Aura/data/`

5.1 Scope

The Project Schedule is intended to provide planning to the level of major activities, such as whole documents. Some detail will be provided for the steps required to complete these activities, but the details are the responsibility of the relevant subteam and development group. The subteams and development groups can plan their own schedule in more detail using the task log system in semester two. Task logs in combination with meeting minutes will provide a good record of detailed schedule planning.

The three main aspects of the project covered by the Project Schedule are:

- Project scheduling and planning;
- Resource requirements;
- Resource allocation.

Project Scheduling and planning is seen to be a most important activity as it allows for the team, subteams and development groups to complete their tasks by a particular date and ensures that the project is completed by the deadline. Contents of the project schedule are outlined below in section 5.2.

Resources are allocated before the start of a phase, or when it is felt that the need arises to allocate team members to particular tasks. The resources are allocated by the management team, following the structure outlined in Section 3.5.2 of this document.

Resource requirements for a particular task, phase, or activity are assigned while scheduling and planning takes place, and gives management an opportunity to estimate resources required for different tasks.

5.2 Contents

The Project Schedule is presented as a Gantt chart. It contains a comprehensive set of information for all the major activities of the project, including:

- A unique name and number for each major activity, and a medium-level breakdown of the sub-activities and tasks making up each activity, in a form that can be "rolled-up" to hide the sub-activity detail;
- Resources required to complete each major activity, including the people involved and a time estimate;
- Project milestones, such as the baselining of major documents;

- An order for the completion of the major activities;
- Interdependencies between different activities, including the critical path;
- A percentage-complete measure for each task.

5.3 Communication and Presentation

The Project Schedule will be put up on the notice board of the 4th year lounge, room L2.04, so all team members can see it at any time conveniently. It will also be referred to regularly at team meetings, subteam leader meetings, and management meetings.

5.4 Project Scheduling Methods

The Project Schedule will initially be evaluated and updated on a weekly basis by the Project Manager after consultation with subteam leaders and members of the management subteam. Subteam leaders will be responsible for informing the Project Manager whether their tasks are fitting to the schedule, and whether any slippage is likely.

Once coding begins, the Project Schedule shall be evaluated and updated by the Technical Leader, who shall perform the same duties which the Project Manager performed in maintaining the Project Schedule. Within development groups, if the development group leader deems it necessary, a development group schedule will be drawn up which should maintain consistency with the Project Schedule.

Once increment two finishes, the Project Schedule will be updated by the manager of the end-of-year MS Access task database.

Estimations for the time required to complete tasks will not be based on any formal metrics. This is because metrics would be meaningless due to the lack of historical data on 440 project timings, the novelty of the project, and because this team has never worked together before.